

|  |
| --- |
| **COURSEWORK ASSESSMENT SPECIFICATION** |

|  |  |
| --- | --- |
| **Module Title:** | *Programming 1* |
| **Module Number:** | *CG0048* |
| **Module Tutor Name(s):** | *Alan Maughan* |
| **Academic Year:** | *20143/15* |
| **% Weighting (to overall module):** | *7.5%* |
| **Coursework Title:** | *Homework 3* |
| **Average Study Time Required by Student:** | *3 hours* |

**Dates and Mechanisms for Assessment Submission and Feedback**

|  |
| --- |
| **Date of Handout to Students:**  Week 7 |
| **Mechanism for Handout to Students:**  *via eLP* |
| **Date and Time of Submission by Student:**  During Week 9 Lab Class |
| **Mechanism for Submission of Work by Student:**  Papers collected in lab |
| **Date by which Work, Feedback and Marks will be returned to Students:**  Marks & feedback will be given as the assessment is marked in the lab week 9. |
| **Mechanism for return of assignment work, feedback and marks to students:**  Marks & feedback will be given as the assessment is marked in the lab |

**Further Information**

*(Please ensure the assessment specification includes the following items)*

**Learning Outcomes tested in this assessment (from the Module Descriptor):**

1. Understand and develop simple Object-Oriented programs.

2. Make use of predefined classes in the development of programs.

**Nature of the submission required:**

Paper copies of source code. Code execution in lab.

**Instructions to students:**

*This is an individual piece of work.*

**Referencing Style:**

*N/A*

**Expected size of the submission**:

Under 12 pages – mostly printed copies of source code

**Academic Conduct:**

You must adhere to the university regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of misconduct or plagiarism in your work. Refer to the University’s regulations on assessment if you are unclear as to the meaning of these terms. The latest copy is available on the university website.

This work is due at the start of your lab in week 9. It counts for 7.5% of the overall module mark.

You must bring with you a printed copy of your source code (the .java files). This should be produced before you come to the lab. Do not come to the lab and attempt to print copies then. These will be collected by the tutor when they mark your homework – make sure that they have your name / id on them. These will be retained for audit and internal moderation. If these files are not submitted (and printed before the lab) then you will score 0 (zero)!

You must work on the program on your own, outside any formal classes and it must be ready to execute at the start of the scheduled laboratory class. All code must be completed using the BlueJ IDE. Any work utilising other IDEs will score zero.

All code (in this and in all subsequent homeworks) must:

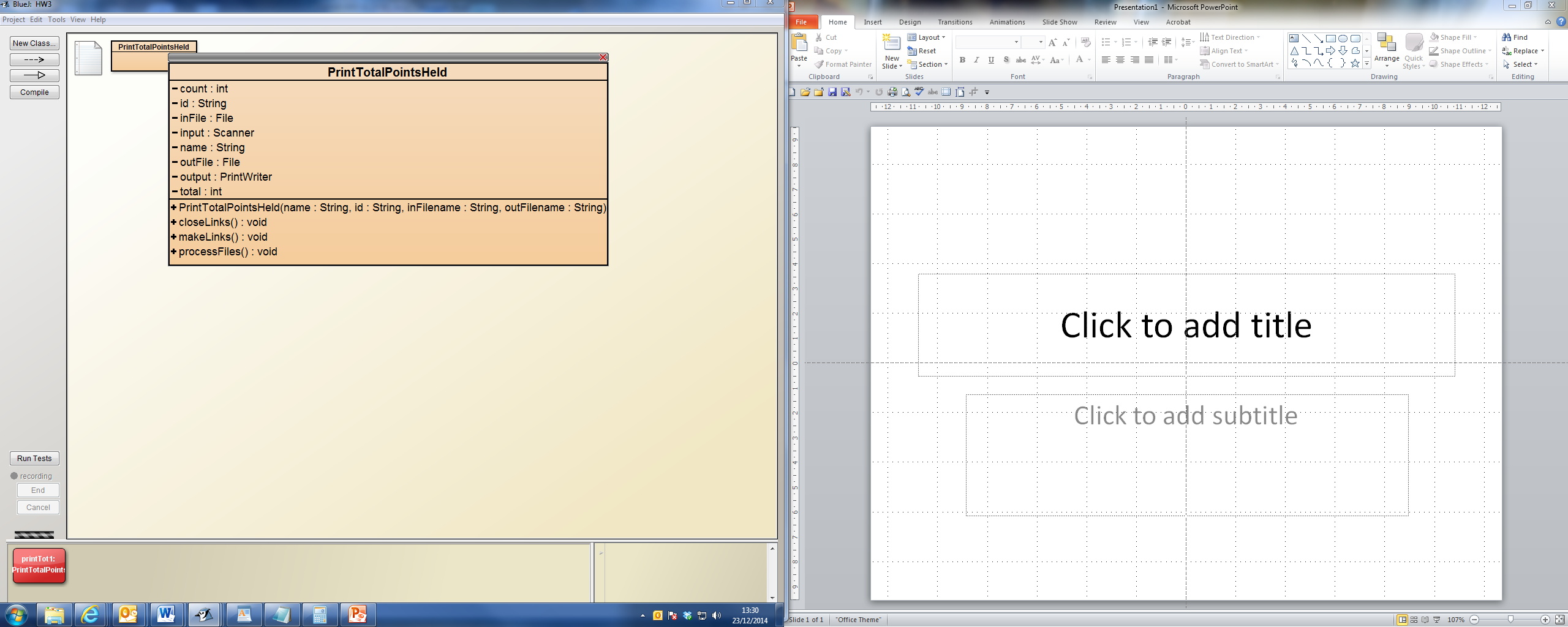
* Have the class header and all methods commented to ‘Javadoc’ standards using @author, @version, @param and @return tags as appropriate. Check one of the classes supplied if you are not certain how these are used.
* Be coded to required layout (e.g. indentation) and naming standards.

Failure to meet these standards will result in loss of a mark..

You may be asked questions about your program to confirm your understanding and that it is your own work. Failure to answer the questions may result in a deduction or total loss of marks.

This homework is intended to test your understanding of exception handling and file I/O. NOTE: This homework includes material from weeks 7 and 8. You may start the homework now but you will not be able to complete it until after then.

For this homework you are required to write a program containing a single class PrintTotalPointsHeld.



When you create an object you will be expected to supply your name and id as strings and two filenames representing text (.txt) files.

* The first filename will be that of a text file (.txt) given to you at the start of the lab. You will need to extract it from BlackBoard at the beginning of your lab and place it in the folder containing your project. You may assume that each line represents a single account and there will be more than one account. It will be in the format:  
    
  firstName lastName accountNumber numberOfPoints

(e.g. Adam Armstrong 1000 23)

* The second file will be that of a text (.txt) file your program must create. You will be given this filename at the start of the lab.

Your program should read the first file and write this information to the second. Whilst processing the first file your program should determine the number of account holders and the total number of books they have on loan and write this to the second file.

The three methods makeLinks(), processFiles() and closeLinks() are expected. You may manually call these methods but for full marks you should automate these calls so that the program runs when the object is created.

The expected text should be:

**Your name and ID**

A Student 12345678

Albert Hammond 1000 111

Bobby Ball 1001 200

Clive Dunn 1003 999

Desmond Dekker 1004 2

Emilio Estevez 1005 0

There are 5 accounts that together hold 1312 points.

**Note:**

**Details of the actual account holders names, the number of account holders and the total number of points will be different.**

Marking Scheme: (5 Marks / 7.5% of module total)

|  |  |
| --- | --- |
| Mark | Criteria |
| 0 | No serious attempt / Program does not compile. |
| 1 | Program compiles. |
| 4 | Text file created.  Data accurate.  Output as expected.  Method calls automated. |